

AMENDMENTS TO THE SPECIFICATION

7, lines 1-17 YR 3/31/2010
Page 5, line 24 to page 6, line 8:

Using clip based data streaming technologies, the user's local computer 120 can play audio or video program in real time as it is being downloaded over the Internet as opposed to pre-storing the entire program in a local file. The Internet radio client application 109 coupled to the web browser 108 decompresses and plays the data as it is being transferred to the local computer 120 over the Internet. The piledriver is responsible for delivering, for example, an Ultravox formatted stream to the client application in a seamless fashion, in addition to raw data. Streaming audio or video avoids the unintended delay entailed in downloading an entire file and then playing it with a helper application. For the clip based streaming to work, the client side receiving the data must be able to collect the data and send it as a steady stream to the program that is processing the data and converting it to sound or pictures. This means that if the data does not come ~~more~~ quickly enough, the presentation of the data will not be smooth. If the streaming client receives the data more quickly than required, it needs to save the excess data in a buffer, which is an area of memory in the write/read random access memory (RAM). Even when the write speed and the read speed are exactly same, to maintain a smooth data flow, a minimum amount of data in the buffer is necessary.

8, lines 12-16 YR 3/31/2010
Page 6, line 28 to page 7, line 2:

There are two reasons to request the pre-buffers in advance. First, it reduces the delay involved in requesting the clip and then obtaining the pre-buffer before being able to play the audio or video. Second, it causes UltraMODS/HTTP to obtain the media file from the content-store if it does not have it already, hopefully in advance of the new request by the client.

10 17-18 YR 3/31/2010
Page 8, lines 15-16:

Call to stop all ~~cache~~ ahead transactions, close and remove all open PFFILEHANDLES and free all used memory.

14 20-21 YR 3/31/2010
Page 11, lines 11-12:

FIG. 3C and FIG. 3D are flow charts further illustrating the various loops according to FIG. 3B.

16 16-19 YR 3/31/2010
Page 13, lines 12-15:

Now No-referring to FIG. 3C, in step 300A, the user's computer checks whether the new target song is already pre-cached by checking whether a file characterized as the new target song exists in the buffer. If not, go to step 331 which includes two sub-steps:

18 7-14 YR 3/31/2010
Page 14, lines 16-23:

The pre-caching (i.e. the pre-buffering) solution described above is possible because the total capacity of the communication channel can be shared between several independent data streams using some kind of multiplexing, in which, each stream's data rate may be limited to a fixed fraction of the total capacity. As it is illustrated in FIG. 4, the data transfer rate for a regular DSL communications channel ranges from 256K to 8M byte per second (bps), the voice conversations and music signals only use 64K bps. Therefore, the streaming track for the downloading data for pre-buffering can use the rest of the capacity.

Amendments to the Specification

0021
Please amend paragraph [0023], as published, of the specification as follows: YR 3/31/2010

From a high level view, the piledriver receives a play list from an audio or video client application. It analyzes the play list and locally caches the first small portion (e.g. first ten seconds) for clip in the play list. The client can then connect to the piledriver data pump and retrieve the data stream using the HTTP or Ultravox 2.0 Protocols. The major functions of the piledriver include: (1) managing the retrieval and caching the ~~pre-buffer~~ pre-buffer for items in the play list; (2) managing the content in memory; (3) providing content to audio or video clients using raw data or the Ultravox 2.0 protocol from either a local cache or directly from a content-store; and (4) providing a stream of data to the audio or video client mimicking local disk functionality.

0057
Please amend paragraph [0068], as published, of the specification as follows: YR 3/31/2010

Referring to FIG. 3C, step 330 actually includes the following two sub-steps:

0070
Please amend paragraph [0081], as published, of the specification as follows: YR 3/31/2010

send request to the server to stop transmitting ~~of the song in~~ the playing song and to start transmitting the new target song; and